

## Transcendental Data

### *Toward a Cultural History and Aesthetics of the New Encoded Discourse*

<preface type="general">

Whether one writes fiction or business reports, prepares lectures or sales presentations, publishes works stored in a library or a commercial database—whatever, in fact, one's domain of authoring might be—the chances are that one is already producing content that somewhere along the route of its transmission takes the form of a uniquely contemporary kind of discourse: encoded or structured discourse, in the technical sense of digital text encoding and structured markup.<sup>1</sup> At its most local, such encoding or markup shows up in the copyedited manuscripts that authors now see from publishers, which, instead of notes to the designer in the old style of "18 pt. heading" (and so on), provide pure logical descriptors or "tags" keyed to house style—for example, the title of this chapter would appear as "<ct>Transcendental

Data</ct>.”<sup>2</sup> More globally, a bewildering variety of the world’s documents and media have in the recent past been encoded in, or are managed by, standardized text-based markup schemes (especially XML, or Extensible Markup Language) that include descriptors for everything from textual or multimedia content to such metadata as author, date, section, and so on. Alternatively, such documents and media have been entered in databases that hold content in tables, records, and fields exportable into XML.

This entire collection of databases and markup languages has so far remained largely hidden from individual writers and readers because it is first being implemented at the institutional level. An increasing number of businesses, publishers, booksellers, university libraries, and digital text archives now use databases and XML to manage the jostling, dynamic bundle of data objects we once called *books, articles, reports, or songs*. But, now that XML is being integrated into standard enterprise and personal-productivity software (including well-known proprietary as well as open-source office suites), ordinary authors and readers—especially those working in institutional settings—will be influenced as well.<sup>3</sup> Authors and readers will join with their institutions to complete a new discursive circuit we might call, updating Friedrich Kittler’s media analysis, *[discourse network 2000]*.<sup>4</sup>

Though the problem of reading the new discourse—browsing, searching, annotating or tagging, and so forth—is intriguing in its own right,<sup>5</sup> I concentrate in this essay on the originating end of the transmission act—authoring. What will discourse network 2000 mean for the act of authoring?

A critical analysis of the new discourse first requires that we unfold its logic in its own terms, which is to say, the technological terms of the postindustrial institutions led by knowledge-work business. But it also requires that we view such techno-logic through the lens of other participating institutions that, while increasingly colonized by postindustrial principles, offer an alternative perspective. The specific institutional perspective I bring to bear is that of the humanities and arts in the academy, which have, in their own way, begun exploring discourse network 2000. Members of the Association for Computers and the Humanities (ACH) and the Text Encoding Initiative (TEI), for example, have long had a hand in developing or adapting text-encoding standards, and, in the last decade, major digital text and image archives at East Coast centers of humanities computing in the United States (such as Brown University and the University of Virginia) have proved the sophistication and robustness of text encoding for literature.<sup>6</sup> In a logically similar move, some of the most advanced humanities computing projects on the U.S. West Coast—for example, those associated with the University of California’s Digital Cultures Project and Digital Arts Research Network (DARnet)—have pursued the

complementary paradigm of database technology.<sup>7</sup> Seen one way, such projects make the transmission of academic knowledge more efficient and flexible and, thus, enroll the humanities and arts in the techno-logic of discourse network 2000. But, viewed differently, they also prepare the academy to refract such techno-logic through its own values, which are not always on the same page with the business master plan. After all, while the technological measure of the new discourse paradigm is postindustrial efficiency coupled with flexibility, that is, the ability to say anything to anyone quickly, the measure of academic knowledge is also historical, social, philosophical, artistic, and public (nonproprietary) diversity, for example, the ability to say anything to anyone fully, richly, openly, differently, kindly, or slowly.

Because one of the emphases in the humanities in the past two decades has been cultural studies, there is special interest at this juncture, I believe, in seeing what a cultural-history approach might tell us about the structured encoding of knowledge.<sup>8</sup> My first critical question will thus be: *What is the social logic that underlies the techno-logic of discourse network 2000?* With specific reference to authoring: *How is an author now a postindustrial producer?* But social history alone is not cultural for the humanities and arts unless it also treats representation, expression, and style, especially as these are now understood to extend beyond the canvas of form onto such subjective and/or material registers of experience as identity or body. My final critical question will open a preliminary speculation into aesthetic logic: *What are the aesthetics of encoded or structured discourse or, as I will term it, of postindustrial dematerialization?* And: *How is it possible for writers or artists to create in such a medium?*

</preface>

<preface type=“technical”>

But, first, what exactly is encoded or structured discourse? Consider the problem of sending a poem over the Internet to a distant computer without knowing exactly what program will receive it, the nature of the processing or display technologies at the other end, or even the remote user’s purpose. The poem is as follows:

The sick ROSE

O Rose thou art sick.

The invisible worm,

That flies in the night  
In the howling storm:

Has found out thy bed  
Of crimson joy:  
And his dark secret love  
Does thy life destroy."

What is the best way for the author to send, not just the content of the poem, but also the exact instructions for processing that content? The general goal is to enable the greatest number of machinic idiots savants at the other end—by turns dumb and brilliant in ways the author cannot predict—not just to receive the poem, but also to do something intelligent with it, whether reproducing the original or something else.

A poor solution, it turns out, is to transmit procedures. A procedural instruction for the display of the poem, for example, might say in essence: "Display the first line beginning at screen position 400 pixels on the x-axis, 500 pixels on the y-axis." Such instructions tell the machine exactly what to do but, for that reason, are not readily adaptable when circumstances vary. (What would a small cell phone screen, e.g., do with a line of text offset 400 pixels to the right?) Procedural instructions also allow for little use of intelligence by the receiving machine to adapt the content to local needs—for instance, to the needs of a hypothetical poem aggregator that looks only for poetry related to disease or written in quatrains (analogous to one of the RSS news aggregator sites that use XML to pull together syndicated news articles from all over the Web).<sup>10</sup> The procedural approach, in other words, addresses only the idiot, not the potential savant that is inhumanly good at filtering, searching, aggregating, transforming, or otherwise processing communication for its own (sometimes unforeseen) purposes.<sup>11</sup>

A better solution, then, is to break the circuit of transmission and reception into two independent parts assisted in their mating by a common standard. On the transmission end, the author can use a logically descriptive rather than a procedural approach to define the elements of the poem—for example, line and stanza—so that they fit within an overall conceptual structure for the parts of a poem. The structure either is implied in a common discursive standard (a set of specifications) that all participants in the discourse network understand or, for customized needs, can be stated explicitly as a set of subordinate, extended standards to be sent along with the content in definitional statements. At the receiving end, the particular program then allows the standard to guide it in using its own procedures

for processing or display. A single description of content at the source can in this way be molded in typically postindustrial fashion to a decentralized variety of consuming programs, formats, and usages.

One common implementation of such an approach is the database solution I previously mentioned. A modern relational and SQL (Structured Query Language) database holds its content descriptively in tables, records, and fields.<sup>12</sup> For example, a set of interrelated tables might contain information about a poem in fields labeled *title*, *author*, *publication\_date*, *line\_number*, and (for the actual content of a particular verse) *line*. Such a database could be "queried" (through a search for *Blake* together with *rose*, e.g.) so as to produce a nicely formatted version of "The Sick Rose." The pure-database approach works best when both the sending and the receiving computers run the same database program, sharing a common discursive standard built into the workings of the software itself. But, for wider compatibility, databases can also present their content on the Web (more precisely, through the Web's HTTP transmission protocol) by means of middleware programming or scripting that shuttles content into HTML (Hypertext Markup Language) or, as we will see below, XML. The rules of HTML or XML, established by the World Wide Web Consortium (W3C), then serve as the common standard that the browser or some other program uses to process the material. A distributed network of users connected by the Web can thus access database content and use Web forms to search or edit that content.

The other major implementation (used by itself or, as suggested above, with databases) is text encoding or markup, whose most current common standard is XML. Developed by the W3C beginning in 1996 as a subset of the older SGML (Standard Generalized Markup Language), XML is designed to be far more capable than HTML (the previous subset of SGML) at sharing information over the Internet in a manner at once uniform and customizable (extensible).<sup>13</sup> XML is human readable in the sense that its descriptive code consists of plain-text tags in angle brackets residing at the same level as the content they encode (i.e., in the same document). These tags, which thus accompany the content wherever it goes, serve the same descriptive function as fields in a database. In XML, for example, our poem might be marked up:

```
<anthology>
  <poem><title>The SICK ROSE</title>
    <stanza>
      <line>O Rose thou art sick.</line>
      <line>The invisible worm,</line>
```

```

<line>That flies in the night</line>
<line>In the howling storm:</line>
</stanza>
<stanza>
  <line>Has found out thy bed</line>
  <line>Of crimson joy:</line>
  <line>And his dark secret love</line>
  <line>Does thy life destroy.</line>
</stanza>
</poem>
</anthology>14

```

Each pair of open and close tags (e.g., `<stanza>` `</stanza>`) describes its enclosed content as an element of the discourse (sometimes supplemented by an attribute or more precise specification of that element). The logical consistency of the whole set of descriptions is guaranteed by a common set of XML rules specified by the W3C such that, to take a simple example, each open tag must be followed by a matching close tag. In addition, the descriptive vocabulary for a particular kind of document—poem, prose, business card, and so on—can be specified in an accompanying DTD (document type definition) or, better, XML schema that explicitly defines, for instance, `<line>` and `<stanza>` so that the former can be nested validly within the latter, but not vice versa. The computer receiving the transmission would then use an XML-processor program working in league with a server application (e.g., a database) or user application (e.g., a browser) to handle the poem appropriately according to the standards.<sup>15</sup>

The overall morphology of discourse network 2000 can thus be outlined as three functionally independent strata, each comprising a set of functions enacted by a variable assemblage of machines, programs, people, and institutional support structures:<sup>16</sup>

*Content Management.* This stratum of discursive activity feeds content into a database or XML source document in structured form, where *structured* means a format able to differentiate and relate such logical units as paragraph, quotation, and title.

*Transmission Management.* This discursive layer exports or sends content over the Internet in the intervening form of XML (together with such supporting standards governing the manipulation or sending of XML as XSLT, SOAP, and so on). Underlying this layer of discourse are the Internet's TCP/IP (Transmission Control Protocol/Internet Protocol) and related

protocols, responsible for transmitting, not documents per se, but their constituent files and data packets.

*Consumption Management.* Consumption management is the stratum of discursive activity that receives the XML transmission and absorbs, reforms, filters, edits, or otherwise *actively* consumes it for local purposes. To use Alvin Toffler's term in his description of postindustrialism, it is productive consumption or "prosumption."<sup>17</sup>

`</preface>`

`<argument title="techno-logic" subtitle="the blind spot on the page">`

On the early world "wild" Web, one corporation's or library's idiosyncratic database might not have been able to send its contents to any other database or end-user program through equally idiosyncratic HTML code. And the gap between ordinary word processing and the Web—let alone the so-called Deep Web of underlying databases—was even wider. But now thick, pliant strands of XML are girding the wilderness (and even tying in word-processor documents) to enable a new order of knowledge. Discourse network 2000 is a step in the direction of what Tim Berners-Lee, the Web's founder, has envisioned as a future "Semantic Web"—a Web that will understand something about the nature of the discourse it is being asked to communicate and, thus, be able to process that discourse more intelligently and automatically.<sup>18</sup>

The techno-logic that informs this vision may be stated in the form of three powerful needs that have converged in contemporary business and other institutions that value the efficient and flexible, which is to say, postindustrial, transmission of information.

The first is the need to make discourse as *transformable* as possible between varying technological and social conventions so that identical content might flow just as easily, for example, to a printed page, a Web page, or a cell phone display.

The second is the need to make discourse *autonomously mobile* in a way that updates Claude Shannon's transmission model of communication (with fixed points of sender and receiver separate from the logic of the message itself). Just as data packets in the Internet's TCP/IP protocol are atomistic parts of a file each with just enough microintelligence about source, destination, and position in the overall file to fly solo like a carrier pigeon and reassemble in proper order with its flock, so in loosely analogous fashion XML elements such as `<line>` and `<stanza>` are what might



be called *document packets* with just enough logical autonomy—assisted by the common XML standard—to know where and how they should be processed relative to the document as a whole. It's like writing a novel on index cards and throwing them out of an airplane at thirty thousand feet. When the cards land, somehow they line up in the right order or—even more uncannily—in someone else's searched, sampled, remixed, summarized, and aggregated order.

The third is the need to *automate* such discourse so that a proliferating population of machinic servers, databases, and client programs can participate as cyborgian agents and concatenated Web services facilitating the processing and reprocessing of knowledge.<sup>19</sup> In the case of an RSS news aggregator, for example, humans no longer need to take many of the intervening steps necessary to find and filter articles residing on different servers and databases. They become just the last of many agents automatically negotiating with each other on the Web to perform the transaction of reading, or browsing, for us.

These cardinal needs of transformability, autonomous mobility, and automation resolve at a more general level into what may be identified as the governing ideology of discourse network 2000: *the separation of content from material instantiation or formal presentation*. Endorsed explicitly or implicitly by the standards-setting bodies (e.g., the TEI, which declares: "XML focuses on the meaning of data, not its presentation"), the ideology of the separation of content from material or formal presentation is the deep logic behind the discursive morphology outlined earlier in which the intervening layer of transmission management serves as something like a secret-agent cutout allowing content management at the source and consumption management at the terminus to be double-blind to each other.<sup>20</sup> From the author's viewpoint, therefore, a poem can now be written free of commitment to, or even knowledge of, the formal or material conventions for receiving the poem.<sup>21</sup> Just as striking is the inverse of the double-blind relation: an author can now create the formal or material conventions for receiving a poem (epitomized in what we now call *interface design*) free of any specific knowledge about what actual source content will be delivered into that frame.

Witness, therefore, the phenomenon of the so-called data island now apparent to Web authors and, with the new generation of XML-based word-processing systems, soon likely to influence the psychology even of mainstream authoring in institutional contexts. Data islands, or, more generally, what I will call *data pours*, are places on a page—whether a Web page or a word-processing page connected live to an institutional database or XML repository—where an author in effect surrenders the act of writ-

ing to that of parameterization.<sup>22</sup> In these topoi, the author designates a zone where content of unknown quantity and quality—except as parameterized in such commands as "twenty items at a time" or "only items containing 'sick rose'"—pours into the manifest work from databases or XML sources hidden in the deep background. Content, in other words, becomes semiotically transcendental (in the sense of what Derrida, in "Structure, Sign, and Play in the Discourse of the Human Sciences," called the "transcendental signified"). The transcendental signified of discourse network 2000 is content that is both the center of discourse and—precisely because of its status as essence—outside the normal play or (as we now say) networking of discourse.

Here are two examples of data islands or data pours from my own past online projects.<sup>23</sup> The first is code from a Web page that draws its content dynamically from a relatively simple database to create a gallery of images for a course on romantic landscape art (figure 8.1).<sup>24</sup> The page initially establishes a connection with the database (liu-images.mdb) and then uses SQL to select all fields from a particular table in that database (the table called *Artists*). Subsequent code on the page requests the content of specific database fields/records and displays it in HTML format for the Web. Records are poured automatically from the database onto the Web page one at a time, but, because the data pour is nested within a repeat statement (which can be variously parameterized), the operation generates a cumulative list or table containing all relevant items in the database even though the author may not know what is in the database or may have ceded control of the database to someone else. The second example represents a data pour implemented through a simple XML document that I created for pedagogical purposes to manage a work of Web-based fiction (Edward Falco's powerful but technically simple *Self-Portrait as Child with Father*), together with an associated XSLT stylesheet (which transforms XML content into some other format, in this case HTML, for the Web). As seen in the XSLT stylesheet, content moves from the XML document to the Web through "select" statements roughly analogous to the select statements used to query SQL databases (figure 8.2).<sup>25</sup>

As a result of such data pours, the interface of the contemporary Web increasingly differs from that of first-generation Web work, where most substantive content was there on the page together with formatting code and other elements under the direct, often idiosyncratic control of the author. Now Web pages increasingly surrender their soul to data pours that throw transcendental information onto the page from database or XML sources reposed far in the background, yet in a manner manifestly different from the thrownness (in Heidegger's phrase) or, more simply,

## Code Excerpt 1

Code from a simple database-driven Web site showing the method by which a Web page calls for content from an underlying database (highlights indicate code discussed in essay):

```
//Connects the Web page to an Access database named "liu-images.mdb,"
//Creates data object through a SQL query "Select" statement

<%
var Recordset1 = Server.CreateObject("ADODB.Recordset");
Recordset1.ActiveConnection = "Provider=Microsoft Jet OLEDB 4.0;
Data Source=E:\databases\liu-images.mdb";
Recordset1.Source = "SELECT * FROM Artists ORDER BY LastName, FirstName, Dates, Nation";
Recordset1.CursorType = 0;
Recordset1.CursorLocation = 2;
Recordset1.LockType = 3;
Recordset1.Open();
var Recordset1_numRows = 0;
%>

//HTML table whose data is pulled through the data connection from records in the database
//The table is set within a Repeat procedure that repeats the process for every record in the database

<% while ((Repeat1__numRows-- != 0) && (!Recordset1.EOF)) { %>
<TABLE width="95%" border="0">
<TR valign="top">
<TD width="40%">
<A HREF="directory-page.asp?<%= MM_keepNone + ((MM_keepNone!="")? "&"&"") + "Artists.ArtistsID="
Recordset1.Fields.Item("ArtistsID").Value %>">
<B><% if (Recordset1.Fields.Item("LastName").Value != "") %>
<%= (Recordset1.Fields.Item("LastName").Value) %>
<% if (Recordset1.Fields.Item("FirstName").Value != "") %>
<%= (Recordset1.Fields.Item("FirstName").Value) %>
<% if (Recordset1.Fields.Item("MiddleName").Value != "") %> &nbsp;
<%= (Recordset1.Fields.Item("MiddleName").Value) %> </B></A>
</TD>
<TD width="20%"><%= (Recordset1.Fields.Item("Dates").Value) %></TD>
<TD width="20%"><%= (Recordset1.Fields.Item("Nation").Value) %></TD>
<TD width="25%"><A href=" "><%= (Recordset1.Fields.Item("InternetSite").Value) %>"> Suggested
Site</A></TD>
<TD width="5%"><%= (Recordset1.Fields.Item("ArtistsID").Value) %></TD>
</TR>
</TABLE>
<% Repeat1__index++; Recordset1.MoveNext(); %>
```

Figure 8.1. Code from a simple database-driven Web site showing the method by which a Web page calls for content from an underlying database (highlights indicate code discussed).

rendered thereness of the rest of the page.<sup>26</sup> I refer to the complex phenomenology of dissonance that appears most visibly, for example, in the telltale way in which data-pour pages eschew “cool” Web design in favor of regular, minimalist, or modernist page layouts with simple geometries (what Lev Manovich calls the “Bauhaus filter”) that is surprisingly prevalent in contemporary information aesthetics.<sup>27</sup> It is only a simpler list or table structure on a Web page, for instance, that can easily receive the serial repetition of an unpredictable number of structurally similar items—that is, the kind of items thrown forth automatically from databases or

## Code Excerpt 2

Code for the XSLT stylesheet “Hypertext\_Fiction.xsl” (from my “Hypertext Fiction Tracker” demo) showing the method by which a Web page calls for content from an underlying XML document. (Highlights indicate code discussed above.)

```
# Called by the XML document "toko_instance.xml" to transform XML into HTML for Web.
# Uses <SPAN> tags to format for HTML the content found in "nodes" in the XML document.
# Note the analogy between the XPATH language "select" statement that locates nodes (e.g.
# "select='REGISTERWORK'" and the SQL "select" statement

<?xml version="1.0"?>
<!-- File Name: Hypertext_Fiction.xsl -->
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">

<xsl:template match="/">

<HTML>
<HEAD>
<TITLE>Hypertext Fiction Tracker</TITLE>
</HEAD>
<BODY>
<SPAN STYLE="font-family:arial, Helvetica, sans-serif; font-size:14px">
<DIV align="center">
<FONT size="4">Hypertext Fiction Tracker</FONT><BR>
<FONT size="3">Conceptual Demo of XML Application</FONT><BR>
<FONT size="2">Alan Liu; last rev. Feb. 18, 2002</FONT>
</DIV><BR>

<xsl:for-each select="REGISTERWORK">
<SPAN STYLE="font-size:130%; font-weight:900">
Title of Work:
<SPAN>
<SPAN STYLE="font-size:130%; font-weight:900; color:red">
<xsl:value-of select="WORK_TITLE"/>
</SPAN><BR>
<SPAN STYLE="font-size:130%; font-weight:900">
Author:
<SPAN>
<SPAN STYLE="font-size:130%; font-weight:900; color:red">
<xsl:value-of select="AUTHOR"/>
</SPAN><BR>
<SPAN STYLE="font-size:130%; font-weight:900">
Date of Publication:
<SPAN>
<SPAN STYLE="font-size:130%; color:red">
<xsl:value-of select="DATE"/>
</SPAN><BR>
<SPAN STYLE="font-size:130%; font-weight:900">
Media Format:
<SPAN>
<SPAN STYLE="font-size:130%; color:red">
<xsl:value-of select="MEDIA"/>
</SPAN><BR>
<SPAN>
<SPAN STYLE="font-size:130%; font-weight:900">
URL:
<SPAN>
<SPAN STYLE="color:red">
<A><xsl:attribute name="href">
<xsl:value-of select="WORK_URL"/>
</xsl:attribute>
<xsl:value-of select="WORK_URL"/>
</A><BR>
<SPAN><BR>
<SPAN STYLE="font-size:130%; font-weight:900; color:red">
Plot Strands:
<SPAN><BR>

<xsl:for-each select="REGISTERWORKPLOT/STRAND">
<SPAN STYLE="font-size:130%; font-weight:900; color:blue">
<xsl:value-of select="STRAND_TITLE"/>
</SPAN><BR><BR>

<xsl:for-each select="LEXIA">
<TABLE width="80%" border="1" align="center" bordercolor:dark: "FFFFFF" bordercolor:"999999">
<TR>
<TD width="15%"><SPAN STYLE="font-weight:800; color:blue">Lexia Title: </SPAN></TD>
<TD><SPAN STYLE="font-weight:800; color:blue">
<xsl:value-of select="LEXIA_TITLE"/></SPAN></TD>
</TR>
<TR>
<TD><SPAN STYLE="font-weight:800">Characters: </SPAN></TD>
<TD><xsl:value-of select="CHARACTERS"/></TD>
</TR>
<TR>
<TD><SPAN STYLE="font-weight:800">Mode: </SPAN></TD>
<TD><xsl:value-of select="MODE"/></TD>
</TR>
<TR>
<TD><SPAN STYLE="font-weight:800">Description or Keywords: </SPAN></TD>
<TD><xsl:value-of select="KEYWORDS"/></TD>
</TR>
<TR>
<TD><SPAN STYLE="font-weight:800">Lexia URL: </SPAN></TD>
<TD><A>
<xsl:attribute name="href"><xsl:value-of select="LEXIA_URL"/></xsl:attribute>
<xsl:value-of select="LEXIA_URL"/>
</A></TD>
</TR>
</TABLE><BR><BR>
<xsl:for-each>
<BR>
<xsl:for-each>
<xsl:for-each>
<SPAN>
</BODY>
</HTML>
</xsl:template>
</xsl:stylesheet>
```

Figure 8.2. Code for the XSLT stylesheet “Hypertext\_Fiction.xsl” (from my “Hypertext Fiction Tracker” demo) showing the method by which a Web page calls for content from an underlying XML document. (Highlights indicate code discussed.)

XML documents that are like volcanoes able to hurl forth only identically shaped rocks.<sup>28</sup> An example would be an RSS news-aggregator site or any of the Internet radio (e.g., SHOUTcast) or streaming/on-demand music sites (e.g., Rhapsody or MusicMatch), where a request for an artist or a kind of music produces a page with a geometrically simple list. In the academic domain, my own *Voice of the Shuttle* throws forth the content of its underlying database in list structures.<sup>29</sup>

In sum, data pours open the prospect of a new model of authoring predicated on technologies enforcing ever more immaculate separation of content from presentation. Or, rather, the term *technology*—along with its whole complement of undecidably objective/social complements (technique, procedure, protocol, routine, practice)—is too narrow. What is at stake is the very *ideology* (as I called it) of strict division between content and presentation—the religion, as it were, of text encoding and databases. Indeed, while I earlier constrained transcendence to semiosis, it would not be inappropriate to inflate semiosis (in the manner of Derrida himself in his discussion of the transcendental signified) to the scale of metaphysics and, in the limit case, religion. Discourse network 2000 is a *belief*.<sup>30</sup> According to its dogma, true content abides in a transcendental logic, reason, or noumenon so completely structured and described that it is in and of itself inutterable in any mere material or instantiated form. Content may be revealed only through an intermediary presentation that is purely interfacial rather than, as it were, sacramental—that is, *not* consubstantial with the noumenal. Unless content is hacked, therefore (which is how our most extreme protestant reformers of information technology today attempt to transcend the interfacial to experience direct revelation), it is to be rendered only through GUIs (graphical user interfaces) that are defined as ipso facto superficial rather than—in the original Orthodox rather than Apple or Microsoft sense—iconic.<sup>31</sup> Unlike an Orthodox icon, which embodies inextricably in its beaten gold the very particles of transcendence, in other words, our interfaces today are ever more transparently just what are termed *skins* or, put technically, *templates*, *schemas*, *stylesheets*, and so on designed to be extricable.

Behold, then: there is now a great blind spot on the page that authors, artists, and designers of the interface no longer directly control but can only parameterize. (Much of the early debate in hypertext theory about the reversal of roles between the author and the newly empowered reader now seems obsolete precisely because *both* the author and the reader are disempowered. Authors and readers become operators of black-box machinery who select criteria for prescribed actions.) In an earlier time, this blind spot through which data floods from transcendental sources might

have been called *the sublime*. Even earlier in the history of transcendence, it was *God*.<sup>32</sup> But now we pray in SQL or XML. Not “Our Father, who art in heaven . . . Give us this day our daily bread,” but, instead, the select statement that is the soul of data islands—for example, “SELECT \* FROM Artists ORDER BY LastName, FirstName, Dates, Nation” (in SQL) or “<xsl:value-of select = “LEXIA\_TITLE” />” (in XML).<sup>33</sup> Not “give us,” in other words, but “select from.” Not the Lord’s Prayer, but our great contemporary prayer, the query.

</argument>

<argument title=“sociologic” subtitle=“rifles, bricks, and forms”>

Of course, the religious analogy is inappropriate to the extent that it misdirects us from the particular church now spreading the discursive Word: postindustrialism. If the principles of transformability, autonomous mobility, and automation that separate content from presentation—and, thus, the juggernaut of databases and XML—currently seem to go without question, then such fatefulness is symptomatic of the exquisitely tight, even supple fit between this rationale and the combined values of industrial efficiency and postindustrial flexibility now responsible for managing our new world order.<sup>34</sup> Such a fit did not arise from above or, what amounts to the same thing, as an entelechy unfolding as progress from the universal reason of humanity. Rather, a cultural-studies approach might show that the alignment of the new discourse with our New Economy is the result of a historical process of *making* things fit. That process we now call *management*, the modern theory of civilization. God begat Enlightenment reason, which begat industrial scientific management, which in turn begat postindustrial management theories that synonymize the progress of civilization and management without any remainder. In the words of Peter Drucker, the scholar who helped found management studies in the United States: “Management . . . converts a mob into an organization, and human efforts into performance.”<sup>35</sup>

My thesis is that the postindustrial techno-logic of encoded or structured discourse dates back—with a signal difference I indicate later—to nineteenth- and early-twentieth-century industrialism. In particular, the mold was set by John Hall and Frederick Winslow Taylor. In regard to Hall, first of all, I am influenced by Wendell Piez, a professional XML developer, theorist of markup languages, and humanist who participates in the ACH.<sup>36</sup> Piez argues that Hall’s now-famous interchangeable-part manufacturing process of the 1820s and 1830s (at the U.S. armory in Harp-



ers Ferry, Virginia) was the predecessor to the logic of separating content from presentation that ultimately triggered not so much databases and XML as the exact social, economic, and technical *need* for databases and XML. We can set the scene by witnessing the increasing complexity of gun manufacture prior to Hall as the [Harpers Ferry armory] ramped up production to meet new, industrial age demands. As described in Merritt Roe Smith's detailed historical study, guns at the armory were still being manufactured as late as 1807 in a pure artisan system. Each craftsman performed all six types of work needed to create a composite product: "barrel making, lock forging, lock filing, brazing, stocking, and finishing."<sup>37</sup> But, when in 1808 the U.S. government dramatically increased its demand to fifteen thousand muskets annually, Harpers Ferry could not muster the necessary number of skilled craftsmen. It thus modified the artisan system by decomposing gun manufacture into separate tasks (soon numbering fifty-five for muskets), each of which could be assigned to a lesser-skilled workman needing to know only, for example, how to make a barrel as opposed to a stock.<sup>38</sup> The composite product was thus dispersed among a network of occupations. Still, the new system left intact the original artisanal method of allowing each worker, no matter how much his task had been simplified, to craft his own gun part. While production numbers went up, therefore, the new system did nothing to prepare for the next demand (in 1815) from the U.S. Ordnance Department: that musket parts be uniform enough so that individual guns could be repaired in the field with parts from other guns made by that same armory and even by the other government armory in Springfield, Massachusetts. Two years of effort dedicated to solving the standardization problem failed to redress what the Ordnance Department called "a total disagreement" between muskets at the two U.S. armories.<sup>39</sup>

It is into this situation that Hall stepped in 1819 when he was appointed the director of the Rifle Works, a new, semi-independent unit at Harpers Ferry designed to manufacture his superior, breech-loading rifle.<sup>40</sup> Importantly, Hall was not just an inventor of the rifle but an indefatigable innovator of the machines, tools, and work processes needed to build rifles. Hall's renovation of the overall system of manufacture ultimately made the difference in the armament industry and U.S. manufacturing as a whole. "At his Rifle Works," Piez observes, "[Hall developed a system by which guns could be made without the hand-crafting traditionally required of them] . . . Instead, the parts were all made to more-than-humanly possible close tolerances by machine, and then assembled not by piece, but by type. That is, any barrel could fit on any stock, with any

receiver, any lock, etc. This required a rigid adherence to standards, enforced by the use of machine tools fitted with jigs, and by a careful regimen of testing with gauges."<sup>41</sup> These "gauges," in fact, were the essence of Hall's new system. Enabling a rigorous method of parts inspection, Hall's case-hardened gauges—distributed in duplicate sets to workmen and to inspectors—soon far outnumbered the inspection devices used for traditional muskets (over sixty-three gauges for one of Hall's breechloader models, e.g.).<sup>42</sup> As Piez argues, in other words, the real proof of quality in Hall's manufacture of a gun was not that the gun fired but that its parts—tested separately in disassembled form—fit against the gauges, which thus became the "Platonic form" of the gun. In the language of XML rather than of Plato, the gauges were the equivalent of a DTD or, better, schema used to validate the particular instance of an XML document against strict standards of complete, consistent, and lawful data structure.<sup>43</sup> "Shades of text-encoding, anyone?" Piez asks.

The crucial point to be made is not that there are particular technical analogies between Hall's rifle manufacture and discourse network 2000, though there are a surprising number of such correspondences—for example, Hall's use of "bearing points" on each rifle part to determine "its relative position for all subsequent machining operations," which might be likened logically to XML namespaces and XPATH nodes that determine, respectively, the bearing of XML tag vocabularies relative to a specific vocabulary set and of the branches of an XML document relative to nodal points in the document structure.<sup>44</sup> The deeper correspondence lies at the level of the overall system of standardization that Hall introduced.<sup>45</sup> While different in the way it creates composite products, postindustrialism starts on the same fundamental requirement of standardization, only its standards are housed, not in gauges, but in an ever more fulsome complement of standards, specifications, DTDs, schemas, and the like.

My own addition to Piez's argument extends the thesis with variation to Taylor, who, at the beginning of the twentieth century, added to Hall's standardized production the management model that takes us a step closer to postindustrialism and what I called *content, transmission, and consumption management*. We might take our example from any of Taylor's case studies of pig-iron handling, shoveling, the manufacture of bicycle bearings, and so on.<sup>46</sup> But perhaps the clearest exemplum is bricklaying as it was studied and reformed by another member of the American Society of Mechanical Engineers, Frank B. Gilbreth, whose work Taylor discusses at length in his *Principles of Scientific Management*. Prior to Gilbreth, bricklayers had built walls in a style akin to that of the traditional armorers we saw



making muskets at Harpers Ferry: they decided ad hoc or by custom how many bricks to cart over, how close to place the pile, how many bricks to lift at one time, how to tamp the bricks down, and so on. But, after reengineering bricklaying (including, e.g., formulating "the exact position which each of the feet of the bricklayer should occupy with relation to the wall, the mortar box, and the pile of bricks"), Gilbreth reduced the motions required to lay each brick from eighteen "to five, and even in one case to as low as two motions."<sup>47</sup> Bricklaying was standardized for efficiency.

Or, rather, standardization was just one of the principles necessary to efficiency in Taylor's system.<sup>48</sup> New in Taylorism was the additional principle that decisions had to be extracted from the embodied work of the laborer and described on instruction cards as procedures that could be optimized, reprogrammed, distributed, and otherwise mediated. The instruction card, Taylor explains in *Shop Management*, specifies, for instance, "the general and detail drawing to refer to, the piece number and the cost order number to charge the work to, the special jigs, fixtures, or tools to use, where to start each cut, the exact depth of each cut, and how many cuts to take, the speed and feed to be used for each cut, and the time within which each operation must be finished."<sup>49</sup> With the introduction of instruction cards (and the Taylorist planning departments that stood behind them), work became the structured, modular, and algorithmically manageable process by which—again translating prophetically into XML—each individual element <BRICK> was nested within the larger element <WALL>. That is, each node or field in the work process (in XML-speak and database-speak, respectively) became part of a programmatic description of wall building that allowed the content (actual bricks lifted by embodied workers) to be separated from the presentation of the wall. Though only superficially akin to modern database forms or XML documents, [Taylor's instruction cards might thus be said to be the first economically and socially significant form of programming—of a piece both logically and chronologically with Herman Hollerith's tabulator punch cards.]

The mediation of work was, in turn, the platform for another principle of efficiency that became Taylor's greatest contribution to industrial history: modern management. In this light, twentieth-century management may be parsed into two correlative ideas. One is that management is management of, and through, media. It is management as document processing or, as JoAnne Yates calls it in her study of document management in the era of Taylor and other systematizers, "control through communication."<sup>50</sup> Once all work decisions were extracted from the laborer and mediated through instruction cards, planning-room diagrams, time-study

charts, and all the other paper apparatuses of Taylor's system, manufacturing could be controlled as if through the revision and rearrangement of documentation alone. The other management idea to emerge in step with mediation is distributed management or what Taylor called "functional foremanship." When manufacturing could be charted out on paper as an interlocking sequence of operations, operators, locations, and resources, then responsibility for the entire plan could be distributed piecemeal to an organization chart of managers that broke the gang-boss mold of management, according to which individual managers directly oversaw platoons of workers. Managers matched up instead with discrete, transposable, and reprogrammable functions that bore no necessary relation to individual workers or work-group formations, which, in turn, could be restructured piecemeal as needed. Workers, in other words, no longer had a boss per se; they were minded instead by a buzzing hive of "order of work and route clerks," "instruction card clerks," "time and cost clerks," "shop disciplinarians," "speed bosses," "inspectors," "repair bosses," and so on who bossed them by bossing around pieces of paper.<sup>51</sup> Freed of the need to be directly bossy, indeed, managers, in Taylor's argument, could even be "friendly," or what we might today call *user-friendly systems of management*.<sup>52</sup> In short, Taylor's functional foremanship was the origin of today's professional-managerial or professional-technical-managerial "new class."<sup>53</sup> Or, perhaps, the real new class to come will not need human managers at all to oversee their user-friendly systems. It is symptomatic that the software client program through which one today administrates a Microsoft SQL Server database is named Enterprise Manager while similar interfaces in the Oracle9i database are called Management Server, Enterprise Management Console, Change Manager, Performance Manager, and so on.<sup>54</sup> Databases and XML are now our ultimate functional managers. They are the automatic mediators of the work of contemporary knowledge.

The missing link between Taylor's paper-pushing managers on the factory floor and today's dedicated document handlers (a.k.a. knowledge workers) in the office can subsequently be found in the books of the Taylorist missionary of clerical and white-collar work William Henry Leffingwell—for example, *Scientific Office Management* (1917) or, with Edwin Marshall Robinson, *Textbook of Office Management* (1932). Leffingwell's time-motion approach to tasks performed on particular kinds of office equipment or in variable lighting and ventilation environments might provide many examples.<sup>55</sup> But the most telling phenomenon is Leffingwell's codification of Taylor's instruction card into the modern "form" (and such related document genres, as Yates calls them, as tables and reports). Near the beginning of their *Textbook of Office Management*, Leffingwell and

Index  
cards.

Robinson compare forms to jigs to show how well-designed documents facilitate the factory-like standardization of office work: "The preparation of a form, for example, was in reality nothing more than the devising of a standard way of recording information. [A form, properly designed, enabled the office manager of that day to get the desired information recorded exactly in the shape he wanted it and without the use of personal instructions. Such a form might be regarded as analogous to the 'jig' used in factories, which enabled a 'machine hand' to perform work which otherwise could have been done only by a skilled mechanic." But the use of standard forms also reinforced the specifically Taylorist innovation of mediated management. As Yates points out, printed forms and tables—especially when they began to converge on a companywide "general logical rule for all forms"—"facilitated the comparisons of data so critical to systematic managers."<sup>57</sup> Leffingwell thus says: "All reports made by one office should follow certain rules devised for that office . . . to the end that all who have occasion to work upon or with the reports, or who use them, may contract uniform and desirable habits of work and thought in relation to this activity."<sup>58</sup> When columns, rows, and fields in one form or table lined up visually with those in other forms, in other words, then not only could information be managed across time and organizational space (e.g., comparing last year's accounting figures against this year's sales figures), but information management became ipso facto worker management as well—a discipline of management powerful enough to be interiorized as a kind of bureaucratic conscience within the psychology of work ("uniform and desirable habits of . . . thought," as Leffingwell says). As Yates observes, organizational communication at the time circulated in a general, upward and downward system of information that helped sustain managerial oversight.<sup>59</sup> Such synoptic management mediated by forms would culminate in the computer era, when, as Shoshana Zuboff studies in *In the Age of the Smart Machine*, the first instinct of both managers and workers on exposure to a firm's computer system was a phenomenology of super-vision. The computer lets them "see" it all.<sup>60</sup>

The upshot of such a social history of databases and XML is that the common presumption of business writers, technologists, and others that there was a sharp break between industrialism and postindustrialism is historically too shallow.<sup>61</sup> There was, indeed, a break, but its distinctive nature cannot be appreciated without first recovering archaeologically the surprising bandwidth of connection between the two epochs. In light of the capitalism underlying both, after all, the separation of content from presentation now being mandated by business-oriented information tech-

nology is a profound euphemism. From a historical perspective, knowledge (the great value of postindustrialism) is being separated or extracted from what presentation really means: labor. What Marx called "surplus labor value" is in the post-Marxist, postindustrial world nothing other than the programmability of work—a programmability that can be functionally managed, extracted, mediated, optimized, and distributed (e.g., licensed to other companies or the end user) for what, in a classically Marxist view, is excess gain.<sup>62</sup>

Premonitions of database- or XML-like features in industrialism are one clue to this congruence between the past and the present. But they are not important in themselves. My historical argument is not one of linear evolution of typological anticipation. Rather, I would deploy a reversed-time variant of the idea of the *skeuomorph* that N. Katherine Hayles uses to explain the history of technological change in *How We Became Posthuman*. Hayles borrows the term *skeuomorph* from archaeological anthropology to describe retro features of the present (like simulated stitching in vinyl molded plastic or, in my context, document forms akin to jigs) that negotiate a comfort zone between the past and the present.<sup>63</sup> Reversing the time arrow, we can say that the database- or XML-like features that we have noticed in the past are a kind of prophetic relic or reverse *skeuomorph*. In their own time, they were proposed as instrumental to the progress of industrialism. But, seen from our perspective, they are epistemological rather than instrumental stitches between past and present. They are an index or a placeholder (rather than a cause or an antecedent) of the future. If such devices as Taylor's instruction cards or Leffingwell's forms (and many other devices that now seem to us merely odd or quaint) did not exactly carry forward to the future, then something else would have been invented to do so—that is, databases and XML.

Only by understanding the deep connection between industrialism and postindustrialism are we now prepared to discern the great difference of the latter. Both epochs, as we have seen, share the projects of standardization and management. But only postindustrialism saw these projects through to their radical conclusion, which might be called *metastandardization and metamanagement*. When Hall standardized the modern rifle, we note, he did so by creating or implementing a host of one-off, stand-alone, ingenious second-order machines for making rifles that were themselves anything but standard. Craft, we might say, merely retreated to the artisanship of Hall himself as the standardizer of craft. So too, when Taylor and Leffingwell brought scientific management to bricklaying or form writing, they only partly regularized the processes of standardization and

management themselves. Some actual human being—much more unpredictable than a machine in aptitudes, skills, and personality—still had to fill the role of the professional-technical manager creating the instruction cards, forms, and so on.<sup>64</sup> The insight of postindustrialism is that there can be metastandards for making standards. XML, for example, is, technically, not a standard, but a metastandard, a family of standards that governs the extensible creation of specific vocabularies of XML tags and their rule sets (such as the TEI guidelines designed for the encoding of scholarly humanities and social-science documents). Similarly, as shown in the massive purge of middle managers in the last few business cycles, postindustrialism is all about the metamanagement of management—the flattening of management layers and the concomitant increase in managerial spans of control made possible by information technologies that transume management through common standards for the transformation, autonomous mobility, and automation of knowledge work. A better name for postindustrialism, we may say, is *metaindustrialism*.

The main point is that a historically deep investigation of discourse network 2000 would inquire into, not just the purpose of data-driven documents (standardization, flexibility, speed, interactivity, collaboration, and so on), but the purpose of that purpose: the foundations of the programmability that today facilitates networked production and consumption. In the context of my specific inquiry, authors are, indeed, postindustrial producers.

Where in the world of rifles, brick walls, and office forms might there still be room for a poem about a sick rose printed in a nonstandard manner and then hand-watercolored by a poet and his wife?

</argument>

<argument title="aesthetics" subtitle="data sublime">

Finally, then, what are the aesthetics of discourse network 2000? Of course, new media studies—which conjoins humanities computing, social-science computing, and the digital arts (including "network art")—has just recently emerged as a coherent, if not wholly unified, field able to address digital, networked communications at the point of intersection between its technology, society, history, philosophy, and aesthetics. Only after ca. 2000, for example, was it possible for new media studies programs to be formed around a more or less shared set of theoretical or other field-defining texts along the lines of books by Espen Aarseth (*Cybertext*), Katherine Hayles

(e.g., *My Mother Was a Computer*), Jerome McGann (e.g., *Radiant Textuality*), Willard McCarty (*Humanities Computing*), and Lev Manovich (*The Language of New Media*), supplemented by an ever richer set of studies by younger scholars marking out new directions of research. (Especially germane among the latter are books such as Matthew G. Kirschenbaum's *Mechanisms* and Rita Raley's *Tactical Media* (and in-progress *Reading Code*) that specifically address the evolution of textual discourse amid the material, code, and interface forms of new media.)<sup>65</sup> Also arriving roughly in the same years were surveys and readers of new media—for example, Noah Wardrip-Fruin and Nick Montfort's important *The New Media Reader*, Stephen Wilson's *Information Arts*, Christiane Paul's *Digital Art*, and the Electronic Literature Organization's *Electronic Literature Collection* series. Taken together, this ensemble of creative, research, and pedagogical materials supporting new media studies can now supply the cross-disciplinary principles and methods that allow us to explore how aesthetics contributes to the total social and historical fact of discourse network 2000.

But in lieu of any attempt at a programmatic synthesis of such principles and methods at the end of this essay, I close simply by mounting a gallery show of past and present art that bears on the theme of data transcendence (what Julian Stallabrass calls the data "sublime") and its impact on the relation between content and presentation.<sup>66</sup> After all, if postindustrial ideology mandates the separation of content from formal presentation or material instantiation, then the arts offer a uniquely critical perspective on this ideology because, in their practice, such separation—however possible as a goal (as demonstrated by some conceptual, digital, or network art)—never goes without saying. The tight, tense marriage between content and materiality/form that the arts witness is at least as powerful as the tight fit between encoded discourse and postindustrialism that divorces content from presentation.

The following is a small gallery of artistic data pours, past and present.

The first instance is a late painting by J. M. W. Turner, *Light and Colour (Goethe's Theory): The Morning after the Deluge* (1843), in which Turner's characteristic vortex of energies, focused on a potent blank spot or white mythology on the canvas, marks a romantic prefiguration of the data pour (figure 8.3). Out of this data pour emerges what seems to be the first record of a transcendental database: an image of Moses writing the Pentateuch. Yet what necessary quantum of aesthetic experience, we may ask, is added by Turner's distinctively formal and material signatures—his rough yet limpid handling of oils and his very imposition of the vortex form?

The second instance is a passage from William Gibson's *Neuromancer*





Figure 8.3. J. M. W. Turner, *Light and Colour (Goethe's Theory): The Morning after the Deluge* (exhibited 1843).

depicting the indescribable, shifting needle of the Kuang virus at the climax of the novel:

Something dark was forming at the core of the Chinese program. The density of information overwhelmed the fabric of the matrix, triggering hypnagogic images. Faint kaleidoscopic angles centered in to a silver-black focal point. Case watched childhood symbols of evil and bad luck tumble out along translucent planes: swastikas, skulls and crossbones, dice flashing snake eyes. If he looked directly at that null point, no outline would form. It took a dozen, quick, peripheral takes before he had it, a shark thing, gleaming like obsidian, the black mirrors of its flanks reflecting faint distant lights that bore no relationship to the matrix around it.<sup>67</sup>

The “null point” of hypnagogic imagery here is the injection point of Gibson’s neuromantic rather than romantic data pour (with strong intimations of Thomas Pynchon in the background).<sup>68</sup> Yet what material and formal instincts—coded, if not in the reptile brain, then at levels of culture far deeper than today’s knowledge work—imagine this null point specifically as swastikas, skulls and crossbones, dice flashing snake eyes, and shark thing?

The third instance is an image by Marcos Novak representative of his “transarchitecture” and “liquid architecture” (figure 8.4). Among his other projects, Novak programs four-dimensional, algorithmic architectural shapes that mutate in time, then extracts from the unimaginable four-dimensional matrix three- or two-dimensional snapshots (through 3D rapid prototyping or 2D imaging). The result is a reduced-dimension data pour from a higher-dimensional reality designed to elicit what Novak terms “allogenic,” truly alien aesthetics beyond the ken even of neuromantic notions of transcendence.<sup>69</sup> Yet what instinct compels Novak to materialize and form his unimaginable 4D shapes in 3D or 2D at all? Why is that stepping down of the dimensional plenum necessary? Moreover, what criteria motivated this *specific* presentation of the unimaginable out

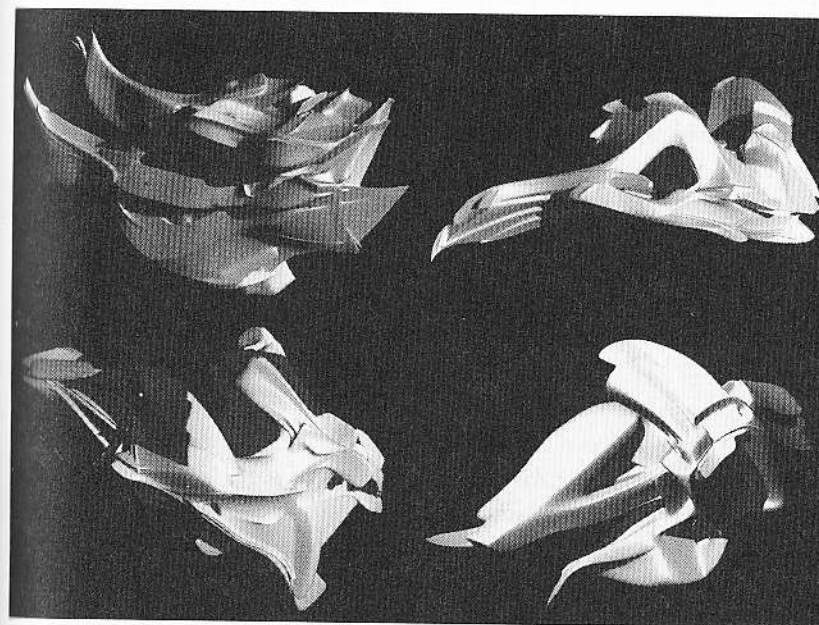


Figure 8.4. Marcos Novak, four views of a four-dimensional transarchitectural shape (2001). Reproduced by permission of the artist.

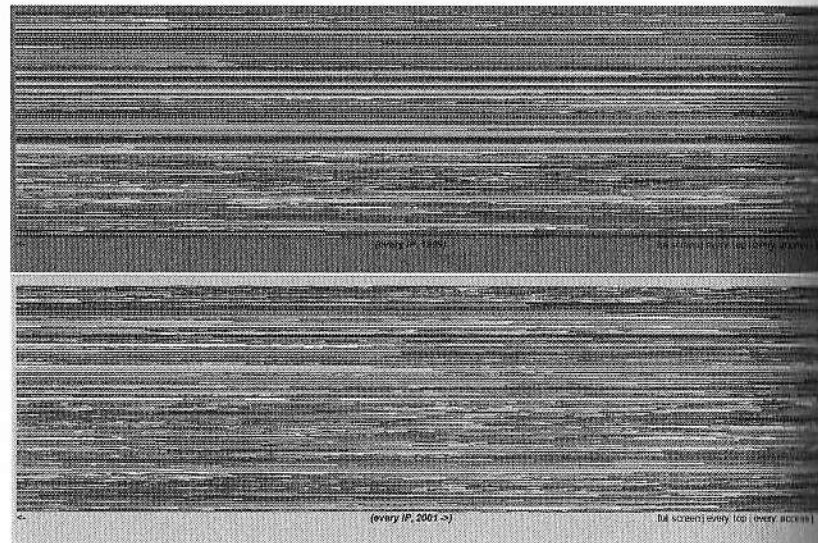


Figure 8.5. Lisa Jevbratt, 1:1, "Every" interface. Reproduced by permission of the artist.

of all the algorithmic possibilities? Why choose this image as especially interesting?

The fourth instance is a now-already-classic work of net art by Lisa Jevbratt that is, perhaps, the epitome of the data sublime. Titled 1:1, the work consists of a net crawler programmed by Jevbratt to access IP addresses on the Internet, a database to hold the addresses together with information about their status (e.g., accessible, inaccessible, returning a server error), and a set of graphical interfaces to display the results while maintaining live, clickable interactivity with the sites behind the IP numbers. The "Every" interface, for example, represents every IP address on the Internet in 1999 and then again in 2001 (figure 8.5).<sup>70</sup> In an astonishing enactment of a data pour, it is a portrait of the Internet in toto (actually, a reduced sampling generated by the crawler) done on an apparently 1:1 scale of correspondence. Yet why did Jevbratt, who started out as a painter, arbitrarily choose particular colors to represent particular aspects of the IP information so as to create a "postphotographic" interface, as the prologue to the project calls it, that may also be appreciated as a recurrence of such painterly postphotographisms as abstract art?<sup>71</sup> Moreover, why did she also make such alternative interfaces for 1:1 as "Migration" (using different colors to track the drift of live addresses on the Internet between 1999 and 2001) that have an even stronger resemblance, not only to the

media-specific form of abstract expressionism, but also to the embodied form of DNA gel or stain sets (figure 8.6)?<sup>72</sup>

Finally, the fifth metainstance is the well-known theoretical prescription from Lyotard's "Answering the Question: What Is Postmodernism?" on the postmodern sublime: "The postmodern would be that which, in the modern, puts forward the unrepresentable in presentation itself; that which denies itself the solace of good forms, the consensus of a taste which would make it possible to share collectively the nostalgia for the unattainable; that which searches for new presentations, not in order to enjoy them but in order to impart a stronger sense of the unrepresentable."<sup>73</sup> The questions we can ask Lyotard in the present context are why "new presentations" are necessary at all and what forms such presentations should take. What is it that makes certain presentations, if not "good," then what Lyotard elsewhere in *The Postmodern Condition* calls "paralogically" interesting?

In the early years of the Web, when data transcendence was presented in piecemeal ways that could still be manually supervised and activated (e.g., hypertext links invoking CGI scripts) or in automated ways that were still relatively crude (e.g., so-called push processes), cool was the dominant aesthetic of the interface.<sup>74</sup> Cool as rendered in HTML, we may say, was the archaic barbarian in the church of the separation of content from presentation. It was the secret adherent of nonstandard, proprietary, hand-coded, and other clearly infidel (or, what is the same in industrial

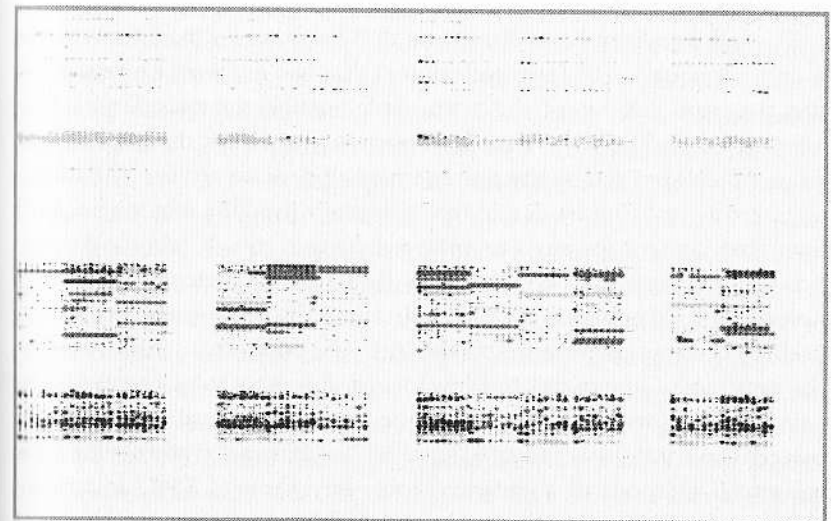


Figure 8.6. Lisa Jevbratt, 1:1, "Migration" interface. Reproduced by permission of the artist.



history, artisanal) practices of embodying content inextricably in presentation (e.g., through pages with so-called dynamic HTML layers that work only in a particular browser or pages with fixed-width tables sized exactly to match a particular graphic image). But, in today's world of massive, automatic data pours through untouchable data islands embedded within retro-Bauhaus rather than cool formalisms (i.e., the regularism of the lists and tables mentioned earlier), what can still be cool?

Here is a first attempt at an analytic of the aesthetics of data transcendence in the age of discourse network 2000. Modern art and literature at the beginning of the twentieth century apprehended the spirit of industrialism in the formalist credo that form is integral to content or, transmuted into the new idiom of process, function. The avant-garde of the time could be famously perverse or obtuse in implementing that formula—offering up everything as art from a Duchamp ready-made to a New Critical verbal icon. Indeed, as I and others have argued, cool in its earlier, Jazz Age usage arose precisely in an orthogonal relation to technological and other functionality.<sup>75</sup> But the avant-garde conviction that there *was* a necessary relation between form and content, nevertheless, reflected industrial standardization and management. Modernists had their gauges too; in their unforgiving manifestos they were faux managers akin to Taylor's shop disciplinarians. Jan Tschichold on the New Typography, for instance, twisted Leffingwellian office forms along the diagonal to stress asymmetry, but the very rigor of his twisted symmetry betrayed that he was still filling out a form (a method later extended into what post-Bauhaus typography called *grid design*).<sup>76</sup>

Postindustrialism, however, shows that there was a third term in the modernist equation of form and content that had not been expressed and that the mere deletion of this term made possible the most far-reaching efficiency-cum-flexibility. That third term is materiality, the implicit substrate of a New Typography poster, imagist poem, or artifact of Bauhaus architecture or furniture design that absolutely fixed the relation between form and content for any one mass-production run. While modernism was the era that first lived with telemedia (telegraphy, telephony, radio), it nevertheless did not grasp the full implications of telepresence. In adapting itself to distributed presence on the Internet, postindustrialism removed the substrate of materiality to allow form and content to be equated in the oxymoronic relation of standard variation or uniform flexibility. When the material substrate was removed to allow for Internet transmission, that is, *variable* methods of standardization—for example, XML documents governed by a common standard but adaptable to undetermined kinds of hardware, software, and usages—could suddenly be imagined. Material

embodiment—in the substrate of a work and the bodily practices of the artisanal artist both—was now immaterial to the full, independent expression of content and form.

Is the writer or artist any longer an author in such circumstances, let alone a creative one? Earlier I said that I would concentrate on authoring in discourse network 2000 as the originating end of the transmission act. The key to the problem of the status of the author, perhaps, lies in the very limitations of that formulation. The origin of transmission in discourse network 2000 is not at the cursor position of the author. Indeed, the heart of the problem of authorship in the age of networked reproduction is that there is no cursor point. We might put the case by updating Kittler's argument about the difference between the 1800 and the 1900 discourse networks in the following fashion. In the romantic era ca. 1800, Kittler observes, the hermeneutical discourse network began when a source of meaning located in nature or the mother called to poets to transmit its transcendental essence through language conceived as a mere channel of translatability. In the modernist era ca. 1900, by contrast, mother nature was a faint echo. The true source of the signal, Kittler argues by recounting psychophysical nonsense reading and speaking experiments of the time, was an apparently random, senseless, automatic, untranslatable, and, thus, nonhermeneutical noise inherent in the channel of transmission itself—like tuning your radio to a Pynchonesque channel of revelation indistinguishable from utter static.<sup>77</sup>

The distinctive signal of 2000, by contrast, synthesizes 1800 and 1900. In 2000, the channel is just as seemingly senseless, random, and automatic as in 1900. (Take a cross section of a document transmission over the Internet at any moment, e.g., and witness a dispersion of atomistic file packets and molar document elements.) But the source point of the transmission is phase-shifted so that phenomenally senseless automatism follows from a precursor act of sense making in the databases and XML repositories outside the direct control of the author. Where the author was once presumed to be the originating transmitter of a discourse next sent for management to the editor, publisher, and so on through all the other positions in the discursive circuit, now the author is in a mediating position as just one among all those other managers looking upstream to previous originating transmitters—database or XML schema designers, software designers, and even clerical information workers (who input data into the database or XML source document). Random and senseless those precursor transmissions may seem (in the way we often feel that overwhelming data is meaningless), yet—in a curious reversion to 1800—that content held in databases and XML now sets the very standard for an ultrastruc-



Mother? tured and ultradescribed rationality purer than any limiting instantiation of the *Ding an Sich*. And so what Kittler calls the “mother’s mouth”—now the discourse of the motherboard, of the matrix itself—seems to return. Only it is alienated from the romantic-era voice of inspiration issuing from the unstructured life that Wordsworth or Blake called “childhood.”

Such a reflection on aesthetics 2000, I think, is the harbinger of a new, but also very old, front for the humanities and arts. The core problem is what I have in my *Laws of Cool* called the “ethos of the unknown”—of the unencoded, unstructured, unmanaged—in human experience. In our current age of knowledge work and total information, what experience of the structurally unknowable can still be conveyed in the structured media of knowledge (databases, XML, and so on)? Perhaps the arts—if they can just crack the code of ordinary cool and make it flower—know.

</argument>

/ / / / / / / / / / /